ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

# Design of Low Complexity Fault Tolerant Parallel FFT Based on Error Correction Codes and Parseval Checks

**Vasam Sathish**
**Department of Electronics & Communication Engineering,**
**DVR College of Engineering and Technology,**
**Hyderabad, Sangareddy District, Telangana, 502283, India.**

## ABSTRACT:

Soft errors pose a reliability threat to modern electronic circuits. This makes protection against soft errors a requirement for many applications. Communications and signal processing systems are no exceptions to this trend. For some applications, an interesting option is to use algorithmic-based fault tolerance (ABFT) techniques that try to exploit the algorithmic properties to detect and correct errors. Signal processing and communication applications are well suited for ABFT.

One example is fast Fourier transforms (FFTs) that are a key building block in many systems. Several protection schemes have been proposed to detect and correct errors in FFTs. Among those, probably the use of the Parseval or sum of squares check is the most widely known. In modern communication systems, it is increasingly common to find several blocks operating in parallel.

Recently, a technique that exploits this fact to implement fault tolerance on parallel filters has been proposed. In this brief, this technique is first applied to protect FFTs. Then, two improved protection schemes that combine the use of error correction codes and Parseval checks are proposed and evaluated. The results show that the proposed schemes can further reduce the implementation cost of protection.

## Keywords:

Error correction codes (ECCs), fast Fourier transforms (FFTs), soft errors.

## I. INTRODUCTION:

The complexity of communications and signal processing circuits increases every year. This is made possible by the CMOS technology scaling that enables the integration of more and more transistors on a single device. This increased complexity makes the circuits more vulnerable to errors. At the same time, the scaling means that transistors operate with lower voltages and are more susceptible to errors caused by noise and manufacturing variations. The importance of radiation-induced soft errors also increases as technology scales. Soft errors can change the logical value of a circuit node creating a temporary error that can affect the system operation.

To ensure that soft errors do not affect the operation of a given circuit, a wide variety of techniques can be used. These include the use of special manufacturing processes for the integrated circuits like, for example, the silicon on insulator. Another option is to design basic circuit blocks or complete design libraries to minimize the probability of soft errors. Finally, it is also possible to add redundancy at the system level to detect and correct errors. One classical example is the use of triple modular redundancy (TMR) that triples a block and votes among the three outputs to detect and correct errors. The main issue with those soft errors mitigation techniques is that they require a large overhead in terms of circuit implementation.

**Cite this article as:** Vasam Sathish, "Design of Low Complexity Fault Tolerant Parallel FFT Based on Error Correction Codes and Parseval Checks", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 4, Issue 12, 2017, Page 175-180.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

For example, for TMR, the overhead is >200%. This is because the unprotected module is replicated three times (which requires a 200% overhead versus the unprotected module), and additionally, voters are needed to correct the errors making the overhead >200%. This overhead is excessive for many applications. Another approach is to try to use the algorithmic properties of the circuit to detect/correct errors. This is commonly referred to as algorithm-based fault tolerance (ABFT). This strategy can reduce the overhead required to protect a circuit.Signal processing and communications circuits are well suited for ABFT as they have regular structures and many algorithmic properties. Over the years, many ABFT techniques have been proposed to protect the basic blocks that are commonly used in those circuits. Several works have considered the protection of digital filters. For example, the use of replication using reduced precision copies of the filter has been proposed as an alternative to TMR but with a lower cost. The knowledge of the distribution of the filter output has also been recently exploited to detect and correct errors with lower overheads.

The protection of fast Fourier transforms (FFTs) has also been widely studied.As signal-processing circuits become more complex, it is common to find several filters or FFTs operating in parallel. This occurs for example in filter banks or in multiple-input multiple-output (MIMO) communication systems [1]. In particular, MIMO orthogonal frequency division modulation (MIMO-OFDM) systems use parallel iFFTs/FFTs for modulation/demodulation. MIMO-OFDM is implemented on long-term evolution mobile systems and also on WiMax[2]. The presence of parallel filters or FFTs creates an opportunity to implement ABFT techniques for the entire group of parallel modules instead of for each one independently. This has been studied for digital filters initially in where two filters were considered. More recently, a general scheme based on the use of error correction codes (ECCs) has been proposed.

In this technique, the idea is that each filter can be the equivalent of a bit in an ECC and parity check bits can be computed using addition. This technique can be used for operations, in which the output of the sum of several inputs is the sum of the individual outputs. This is true for any linear operation as, for example, the discrete Fourier transform (DFT).In this brief, the protection of parallel FFTs is studied. In particular, it is assumed that there can only be a single error on the system at any given point in time. This is a common assumption when considering the protection against radiation-induced soft errors. There are three main contributions in this brief.

1) The evaluation of the ECC technique for the protection of parallel FFTs showing its effectiveness in terms of overhead and protection effectiveness.
2) The proposal of a new technique based on the use of Parseval or sum of squares (SOSs) checks [4] combined with a parity FFT.
3) The proposal of a new technique on which the ECC is used on the SOS checks instead of on the FFTs.

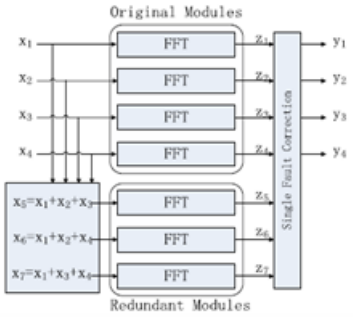



**Fig.Parallel FFT protection using ECCs.**

The two proposed techniques provide new alternatives to protect parallel FFTs that can be more efficient than protecting each of the FFTs independently. The proposed schemes have been evaluated using FPGA implementations to assess the protection overhead. The results show that by combining the use of ECCs and Parsevalchecks, the protection overhead can be reduced compared with the use of only ECCs as proposed. Fault injection experiments have also been conducted to verify the ability of the implementations to detect and correct errors.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The rest of this brief is organized as follows. Section II presents the two proposed schemes. In Section III, the implementation over-heads and fault tolerance of the schemes are evaluated. Finally, the conclusions are drawn in Section IV.

## II. PROPOSED PROTECTION SCHEMES FOR PARALLEL FFTS:

The starting point for our work is the protection scheme based on the use of ECCs that was presented for digital filters. This scheme is shown in Fig. 1. In this example, a simple single error correction Hamming code is used. The original system consists of four FFT modules and three redundant modules is added to detect and correct errors. The inputs to the three redundant modules are linear combinations of the inputs and they are used to check linear combinations of the outputs. For example, the input to the first redundant module is

$$x_5 = x_1 + x_2 + x_3 \qquad (1)$$

and since the DFT is a linear operation, its output $z_5$ can be used to check that

$$z_5 = z_1 + z_2 + z_3. \qquad (2)$$

This will be denoted as $c_1$ check. The same reasoning applies to the other two redundant modules that will provide checks $c_2$ and $c_3$. Based on the differences observed on each of the checks, the module on which the error has occurred can be determined. The different patterns and the corresponding errors are summarized in Table I. Once the module in error is known, the error can be corrected by reconstructing its output using the remaining modules. For example, for an error affecting $z_1$, this can be done as follows:

$$z_{1c}[n] = z_5[n] - z_2[n] - z_3[n]. \qquad (3)$$

Similar correction equations can be used to correct errors on the other modules. More advanced ECCs can be used to correct errors on multiple modules if that is needed in a given application. The overhead of this technique, as discussed , is lower than TMR as the number of redundant FFTs is related to the logarithm of the number of original FFTs.

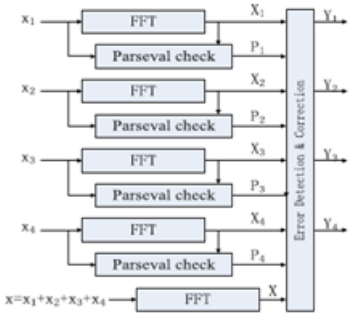



**Fig. 2. Parity-SOS (first technique) fault-tolerant parallel FFTs.**

of the number of original FFTs For example, to protect four FFTs, three redundant FFTs are needed, but to protect eleven, the number of redundant FFTs in only four. This shows how the overhead decreases with the number of FFTs.In Section I, it has been mentioned that over the years, many techniques have been proposed to protect the FFT. One of them is the Sum of Squares (SOSs) check [4] that can be used to detect errors. The SOS check is based on the Parseval theorem that states that the SOSs of the inputs to the FFT are equal to the SOSs of the outputs of the FFT except for a scaling factor. This relationship can be used to detect errors with low overhead as one multiplication is needed for each input or output sample (two multiplications and adders for SOS per sample).

For parallel FFTs, the SOS check can be combined with the ECC approach to reduce the protection overhead. Since the SOS check can only detect errors, the ECC part should be able to implement the correction. This can be done using the equivalent of a simple parity bit for all the FFTs. In addition, the SOS check is used on each FFT to detect errors. When an error is detected, the output of the parity FFT can be used to correct the error. This is better explained with an example. In Fig. 2, the first proposed scheme is illustrated for the case of four parallel FFTs. A redundant (the parity) FFT is added that has the sum of the inputs to the original FFTs as input. An SOS check is also added to each original FFT.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In case an error is detected (using $P_1$, $P_2$, $P_3$, $P_4$), the correction can be done by recomputing the FFT in error using the output of the parity FFT (X ) and the rest of the FFT outputs. For example, if an error occurs in the first FFT, $P_1$ will be set and the error can be corrected by doing

$$X_{1c} = X - X_2 - X_3 - X_4. \qquad (4)$$

This combination of a parity FFT and the SOS check reduces the number of additional FFTs to just one and may, therefore, reduce the protection overhead. In the following, this scheme will be referred to as parity-SOS (or first proposed technique).Another possibility to combine the SOS check and the ECC approach is instead of using an SOS check per FFT, use an

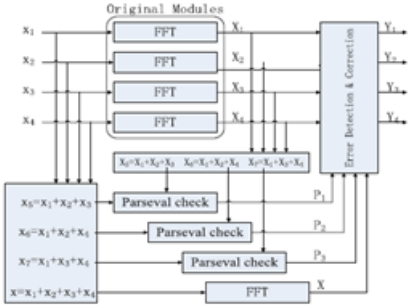



**Fig.Parity-SOS-ECC (second technique) fault-tolerant parallel FFTs.**

ECC for the SOS checks. Then as in the parity-SOS scheme, an additional parity FFT is used to correct the errors. This second technique is shown in Fig. 3. The main benefit over the first parity-SOS scheme is to reduce the number of SOS checks needed. The error location process is the same as for the ECC scheme in Fig. 1 and correction is as in the parity-SOS scheme. In the following, this scheme will be referred to as parity-SOS-ECC (or second proposed technique).The overheads of the two proposed schemes can be initially estimated using the number of additional FFTs and SOS check blocks needed. This information is summarized in Table II for a set of k original FFT modules assuming k is a power of two. It can be observed that the two proposed schemes reduce the number of additional FFTs to just one.

In addition, the second technique also reduces the number of SOS checks. In Section III, a detailed evaluation for an FPGA implementation is discussed to illustrate the relative overheads of the proposed techniques.In all the techniques discussed, soft errors can also affect the elements added for protection. For the ECC technique[5], the protection of these elements was discussed. In the case of the redundant or parity FFTs, an error will have no effect as it will not propagate to the data outputs and will not trigger a correction.

In the case of SOS checks, an error will trigger a correction when actually there is no error on the FFT. This will cause an unnecessary correction but will also produce the correct result. Finally, errors on the detection and correction blocks in Figs. 2 and 3 can propagate errors to the outputs. In our implementations, those blocks are protected with TMR. The same applies for the adders used to compute the inputs to the redundant FFTs in Fig. 1 or to the SOS checks in Fig. 3. The triplication of these blocks has a small impact on circuit complexity as they are much simpler than the FFT computations.

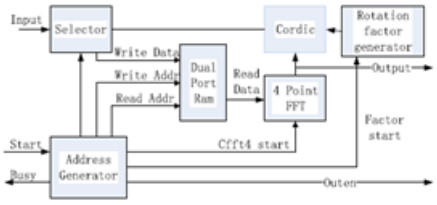



**Fig. Architecture of the FFT implementation**

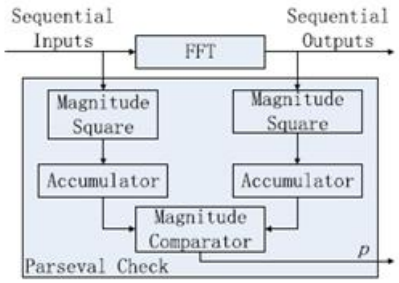
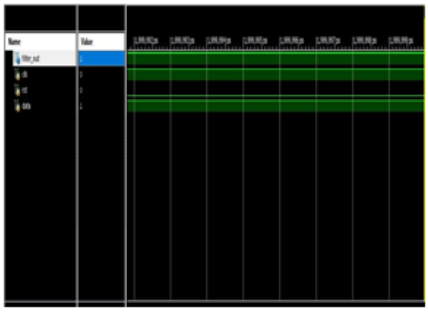


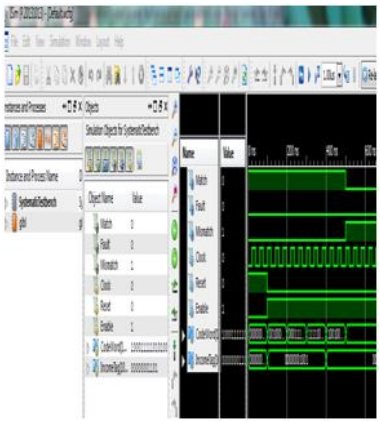

**Fig. Implementation of the SOS check.**

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A final observation is that the ECC scheme can detect all errors that exceed a given threshold (given by the quantization used to implement the FFTs). On the other hand, the SOS check detects most errors but does not guarantee the detection of all errors [6]. Therefore, to compare the three techniques for a given implementation, fault injection experiments should be done to determine the percentage of errors that are actually corrected. This means that an evaluation has to be done both in terms of overhead and error coverage.

## III.RESULTS:





### Comparison:

| Existing Technique | Proposed Technique |
|---|---|
| Error is detected | Error is corrected |
| Half adder is used. | Reversible Half adder is used to produce unique output pattern. |
| Device Utilisation-154 components | Device Utilisation-124 components |
| Time Utilisation-32.476ns | Time Utilisation-24.233ns |

## V.CONCLUSION:

In this brief, the protection of parallel FFTs implementation against soft errors has been studied. Two techniques have been proposed and evaluated. The proposed techniques are based on combining an existing ECC approach with the traditional SOS check. The SOS checks are used to detect and locate the errors and a simple parity FFT is used for correction. The detection and location of the errors can be done using an SOS check per FFT or alternatively using a set of SOS checks that form an ECC. The proposed techniques have been evaluated both in terms of implementation complexity and error detection capabilities. The results show that the second technique, which uses a parity FFT and a set of SOS checks that form an ECC, provides the best results in terms of implementation complexity. In terms of error protection, fault injection experiments show that the ECC scheme can recover all the errors that are out of the tolerance range. Thefault coverage for the parity-SOS scheme and the parity-SOS-ECC scheme is ~99.9% when the tolerance level for SOS check is 1.

## REFERENCES:

[1] N. Kanekawa, E. H. Ibe, T. Suga, and Y. Uematsu, Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances. New York, NY, USA: Springer-Verlag, 2010.

[2] R. Baumann, "Soft errors in advanced computer systems," IEEE Des. Test Comput., vol. 22, no. 3, pp. 258–266, May/Jun. 2005.

[3] M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.

[4] A. L. N. Reddy and P. Banerjee, "Algorithm-based fault detection for signal processing applications," IEEE Trans. Comput., vol. 39, no. 10, PP.1304-1308, Oct. 1990.

[5] T. Hitana and A. K. Deb, "Bridging concurrent and non-concurrent error detection in FIR filters," in Proc. Norchip Conf., Nov. 2004.

[6] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, "Totally fault tolerant RNS based FIR filters," in Proc. 14th IEEE Int. On-Line Test Symp. (IOLTS), Jul. 2008, pp. 192–194.

**Author Details:**

**Vasam Sathish**

He received B.Tech Degree in Electrical & Electronics Engineering from Trinity College of Engineering and Technology Karimnagar India. Presently pursuing M.Tech (VLSI & EMBEDDED SYSTEMS) from DVR College of Engineering And Technology, Hyderabad, Sangareddy District, Telangana, India.